CLAIMS

What is claimed is:

1. A data dependent scrambler for a communications channel that receives a user data sequence including N symbols each with M bits, comprising:

a seed finder that selects a scrambling seed;

a first scrambler that receives said user data sequence and said scrambling seed from said seed finder and that generates a scrambled user data sequence; and

a first encoder that identifies a string of X consecutive zeros in adjacent symbols of said scrambled used data sequence, that replaces one of said adjacent symbols with an all-one symbol, and that replaces the other of said adjacent symbols with first bits representing a position of said string of X consecutive zeros and second bits representing bits of said adjacent symbols that are not in said string of X consecutive zeros.

2. The data dependent scrambler of Claim 1 wherein $X > M$.

3. The data dependent scrambler of Claim 1 wherein both of said symbols are not all-zero.

4. The data dependent scrambler of Claim 1 wherein said seed finder selects said scrambling seed based upon said symbols in said user data sequence.

5. The data dependent scrambler of Claim 1 wherein X = 12 and M=10.

6. The data dependent scrambler of Claim 1 wherein said start position is identified using Y least significant bits of said second symbol.

7. The data dependent scrambler of Claim 6 wherein Y = 3.

8. The data dependent scrambler of Claim 1 further comprising:

an H-code token finder that generates an H-code token that is dependent upon said symbols in said user data sequence; and

an H-code encoder that receives said scrambled user data sequence from said scrambler, that increases a Hamming weight of said scrambled user data sequence using said H-code token and that outputs said scrambled user data to said encoder.

9. The data dependent scrambler of Claim 1 wherein said data dependent scrambler is implemented in a write path of a data storage system.

10. The data dependent scrambler of Claim 1 wherein said scrambler performs a bitwise exclusive (XOR) operation.

11. A communications channel that receives a user data sequence including N symbols each with M bits, comprising:

a host bus interface (HBI) that receives said user data sequence;

a data dependent scrambler that scrambles said user data sequence, including:

a seed finder that selects a scrambling seed;

a first scrambler that receives said user data sequence and said scrambling seed from said seed finder and that generates a scrambled user data sequence; and

a first encoder that identifies a string of X consecutive zeros in adjacent symbols of said scrambled used data sequence, that replaces one of said adjacent symbols with an all-one symbol, and that replaces the other of said adjacent symbols with first bits representing a position of said string of X consecutive zeros and second bits representing bits of said adjacent symbols that are not in said string of X consecutive zeros.

12. The communications channel of Claim 11 wherein $X > M$.

13. The communications channel of Claim 11 wherein both of said symbols are not all-zero.

14. The communications channel of Claim 11 wherein said seed finder selects said scrambling seed based upon said symbols in said user data sequence.

15. The communications channel of Claim 11 wherein X = 12 and M=10.

16. The communications channel of Claim 11 wherein said start position is identified using Y least significant bits of said second symbol.

17. The communications channel of Claim 16 wherein Y = 3.

18. The communications channel of Claim 11 wherein said data dependent scrambler further comprises:

an H-code token finder that generates an H-code token that is dependent upon said symbols in said user data sequence; and

an H-code encoder that receives said scrambled user data sequence from said scrambler, that increases a Hamming weight of said scrambled user data sequence using said H-code token and that outputs said scrambled user data to said encoder.

19. The communications channel of Claim 11 wherein said communications channel is a write path of a data storage system.

20. The communications channel of Claim 11 wherein said scrambler performs a bitwise exclusive (XOR) operation.

21. The communications channel of Claim 11 further comprising an error correction coding (ECC) and cyclical redundancy check (CRC) encoder that encodes said scrambled user data sequence and appends ECC and CRC bits to said scrambled user data sequence.

22. The communications channel of claim 21 further comprising a run length limited (RLL) encoder that encodes said ECC and CRC bits and appends RLL bits to said scrambled user data sequence.

23. A data dependent scrambler for a communications channel that receives a user data sequence including N symbols each with M bits, comprising:

seed finding means for selecting a scrambling seed;

first scrambling means for receiving said user data sequence and said scrambling seed and for generating a scrambled user data sequence; and

first encoding means for identifying a string of X consecutive zeros in adjacent symbols of said scrambled used data sequence, for replacing one of said adjacent symbols with an all-one symbol, and for replacing the other of said adjacent symbols with first bits representing a position of said string of X consecutive zeros and second bits representing bits of said adjacent symbols that are not in said string of X consecutive zeros.

24. The data dependent scrambler of Claim 23 wherein $X > M$.

25. The data dependent scrambler of Claim 23 wherein both of said symbols are not all-zero.

26. The data dependent scrambler of Claim 23 wherein said seed finding means selects said scrambling seed based upon said symbols in said user data sequence.

27. The data dependent scrambler of Claim 23 wherein $X = 12$ and $M=10$.

28. The data dependent scrambler of Claim 23 wherein said start position is identified using Y least significant bits of said second symbol.

29. The data dependent scrambler of Claim 28 wherein Y = 3.

30. The data dependent scrambler of Claim 23 further comprising:

token finding means for generating a token that is dependent upon said symbols in said user data sequence; and

second encoding means for receiving said scrambled user data sequence from said scrambling means, for increasing a Hamming weight of said scrambled user data sequence using said token and for outputting said scrambled user data to said first encoding means.

31. The data dependent scrambler of Claim 23 wherein said data dependent scrambler is implemented in a write path of a data storage system.

32. The data dependent scrambler of Claim 23 wherein said scrambling means performs a bitwise exclusive (XOR) operation.

33. A communications channel that receives a user data sequence including N symbols each with M bits, comprising:

interface means for receiving said user data sequence;

data dependent scrambling means for scrambling said user data sequence, including:

seed finding means for selecting a scrambling seed;

first scrambling means for receiving said user data sequence and said scrambling seed from said seed finder and for generating a scrambled user data sequence; and

first encoding means for identifying a string of X consecutive zeros in adjacent symbols of said scrambled used data sequence, for replacing one of said adjacent symbols with an all-one symbol, and for replacing the other of said adjacent symbols with first bits representing a position of said string of X consecutive zeros and second bits representing bits of said adjacent symbols that are not in said string of X consecutive zeros.

34. The communications channel of Claim 33 wherein $X > M$.

35. The communications channel of Claim 33 wherein both of said symbols are not all-zero.

36. The communications channel of Claim 33 wherein said seed finding means selects said scrambling seed based upon said symbols in said user data sequence.

37. The communications channel of Claim 33 wherein X = 12 and M=10.

38. The communications channel of Claim 33 wherein said start position is identified using Y least significant bits of said second symbol.

39. The communications channel of Claim 38 wherein Y = 3.

40. The communications channel of Claim 33 wherein said data dependent scrambling means further comprises:

token finding means for generating a token that is dependent upon said symbols in said user data sequence; and

second encoding means for receiving said scrambled user data sequence from said scrambler, for increasing a Hamming weight of said scrambled user data sequence using said token and for outputting said scrambled user data to said first encoding means.

41. The communications channel of Claim 33 wherein said communications channel is a write path of a data storage system.

42. The communications channel of Claim 33 wherein said scrambling means performs a bitwise exclusive (XOR) operation.

43. The communications channel of Claim 33 further comprising error correction coding (ECC) and cyclical redundancy check (CRC) encoding means for encoding said scrambled user data sequence and for appending ECC and CRC bits to said scrambled user data sequence.

44. The communications channel of claim 43 further comprising run length limited (RLL) encoding means for encoding said ECC and CRC bits and for appending RLL bits to said scrambled user data sequence.

45. A method of scrambling a user data sequence including N symbols each with M bits, comprising:

selecting a scrambling seed;

generating a scrambled user data sequence based on said user data sequence and said scrambling seed;

identifying a string of X consecutive zeros in adjacent symbols of said scrambled used data sequence;

replacing one of said adjacent symbols with an all-one symbol; and

replacing the other of said adjacent symbols with first bits representing a position of said string of X consecutive zeros and second bits representing bits of said adjacent symbols that are not in said string of X consecutive zeros.

46. The method of Claim 45 wherein $X > M$.

47. The method of Claim 45 wherein both of said symbols are not all-zero.

48. The method of Claim 45 wherein said scrambling seed is based upon said symbols in said user data sequence.

49. The method of Claim 45 wherein $X = 12$ and $M=10$.

50. The method of Claim 45 further comprising identifying said start position using Y least significant bits of said second symbol.

51. The method of Claim 50 wherein Y = 3.

52. The method of Claim 45 further comprising:

generating an H-code token that is dependent upon said symbols in said user data sequence; and

increasing a Hamming weight of said scrambled user data sequence using said H-code token.

53. The method of Claim 45 wherein said step of scrambling includes performing a bitwise exclusive (XOR) operation.

54. A data dependent descrambler for a communications channel that receives a scrambling seed and a scrambled user data sequence with N symbols each with M bits, comprising:

a first decoder that analyzes adjacent symbols of said scrambled user data sequence, that performs G-constraint decoding on said adjacent symbols when a first of said adjacent symbols is an all-one symbol and that does not perform G-constraint encoding when said first of said adjacent symbols is not an all-one symbol; and

a first descrambler that communicates with said first decoder and that generates a user data sequence based on said scrambled user data sequence and said scrambling seed.

55. The data dependent descrambler of Claim 54 wherein said first decoder performs G-constraint decoding by replacing said all-one symbol with an all-zero symbol, by identifying a start position of a string of X consecutive zeros in said adjacent symbols of said scrambled used data sequence, by recovering second bits from said adjacent symbols representing bits of said adjacent symbols that are not in said string of X consecutive zeros, and by substituting said recovered second bits into said adjacent symbols.

56. The data dependent descrambler of Claim 54 wherein $X > M$.

57. The data dependent descrambler of Claim 56 wherein X = 12 and M=10.

58. The data dependent descrambler of Claim 54 wherein said start position is identified using Y least significant bits of a second one of said adjacent symbols.

59. The data dependent descrambler of Claim 58 wherein Y = 3.

60. The data dependent descrambler of Claim 54 further comprising an H-code decoder that receives said scrambled user data sequence from said first decoder, that decreases a Hamming weight of said scrambled user data sequence using an H-code token and that outputs said scrambled user data to said descrambler.

61. The data dependent descrambler of Claim 54 wherein said data dependent descrambler is implemented in a write path of a data storage system.

62. The data dependent descrambler of Claim 54 wherein said descrambler performs a bitwise exclusive (XOR) operation.

63. A data dependent descrambler for a communications channel that receives a scrambling seed and a scrambled user data sequence with N symbols each with M bits, comprising:

first decoding means for analyzing adjacent symbols of said scrambled user data sequence, for performing G-constraint decoding on said adjacent symbols when a first of said adjacent symbols is an all-one symbol and for not performing G-constraint encoding when said first of said adjacent symbols is not an all-one symbol; and

first descrambling means that communicates with said first decoding means for generating a user data sequence based on said scrambled user data sequence and said scrambling seed.

64. The data dependent descrambler of Claim 63 wherein said first decoding means performs G-constraint decoding by replacing said all-one symbol with an all-zero symbol, by identifying a start position of a string of X consecutive zeros in said adjacent symbols of said scrambled used data sequence, by recovering second bits from said adjacent symbols representing bits of said adjacent symbols that are not in said string of X consecutive zeros, and by substituting said recovered second bits into said adjacent symbols.

65. The data dependent descrambler of Claim 63 wherein $X > M$.

66. The data dependent descrambler of Claim 65 wherein X = 12 and M=10.

67. The data dependent descrambler of Claim 63 wherein said start position is identified using Y least significant bits of a second one of said adjacent symbols.

68. The data dependent descrambler of Claim 67 wherein Y = 3.

69. The data dependent descrambler of Claim 63 further comprising H-code decoding means for receiving said scrambled user data sequence from said first decoder, for decreasing a Hamming weight of said scrambled user data sequence using an H-code token and for outputting said scrambled user data to said descrambling means.

70. The data dependent descrambler of Claim 63 wherein said data dependent descrambler is implemented in a write path of a data storage system.

71. The data dependent descrambler of Claim 63 wherein said descrambling means performs a bitwise exclusive (XOR) operation.

72. A method for operating a data dependent descrambler for a communications channel that receives a scrambling seed and a scrambled user data sequence with N symbols each with M bits, comprising:

analyzing adjacent symbols of said scrambled user data sequence;

performing G-constraint decoding on said adjacent symbols when a first of said adjacent symbols is an all-one symbol; and

not performing G-constraint encoding when said first of said adjacent symbol is not an all-one symbol.

73. The method of Claim 72 further comprising generating a user data sequence based on said scrambled user data sequence and said scrambling seed.

74. The method of Claim 72 wherein said performing G-constraint decoding step includes:

replacing said all-one symbol with an all-zero symbol;

identifying a start position of a string of X consecutive zeros in said adjacent symbols of said scrambled used data sequence;

recovering second bits from said adjacent symbols representing bits of said adjacent symbols that are not in said string of X consecutive zeros, and

substituting said recovered second bits into said adjacent symbols.

75. The method of Claim 72 wherein X > M.

76. The method of Claim 74 wherein X = 12 and M=10.

77. The method of Claim 72 further comprising identifying said start position using Y least significant bits of a second one of said adjacent symbols.

78. The method of Claim 77 wherein Y = 3.

79. The method of Claim 72 further comprising decreasing a Hamming weight of said scrambled user data sequence using an H-code token.